



Li, L., Oikonomou, G., Beach, M., Nejabati, R., & Simeonidou, D. (2019). An SDN Agent-Enabled Rate Adaptation Framework for WLAN. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)* [8761424] (IEEE International Conference on Communications; Vol. 2019-May). Institute of Electrical and Electronics Engineers (IEEE).  
<https://doi.org/10.1109/ICC.2019.8761424>

Peer reviewed version

License (if available):  
Other

Link to published version (if available):  
[10.1109/ICC.2019.8761424](https://doi.org/10.1109/ICC.2019.8761424)

[Link to publication record in Explore Bristol Research](#)  
PDF-document

This is the accepted author manuscript (AAM). The final published version (version of record) is available online via IEEE at <https://doi.org/10.1109/ICC.2019.8761424> . Please refer to any applicable terms of use of the publisher.

## University of Bristol - Explore Bristol Research

### General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:  
<http://www.bristol.ac.uk/pure/about/ebr-terms>

# An SDN Agent-enabled Rate Adaptation Framework for WLAN

Li Li, George Oikonomou, Mark Beach  
Communications Systems and Networks Group  
University of Bristol  
Bristol, United Kingdom  
[l.li, g.oikonomou, m.a.beach}@bristol.ac.uk](mailto:{l.li, g.oikonomou, m.a.beach}@bristol.ac.uk)

Reza Nejabati, Dimitra Simeonidou  
High Performance Networks Group  
University of Bristol  
Bristol, United Kingdom  
[reza.nejabati, dimitra.simeonidou}@bristol.ac.uk](mailto:{reza.nejabati, dimitra.simeonidou}@bristol.ac.uk)

**Abstract**— Rate or link adaptation is the determination of the optimal modulation and coding scheme (MCS) that will maximize the performance under the current wireless channel conditions. A Software-Defined Networking (SDN) agent is a software element bridging an SDN controller and any legacy wireless network elements by providing the abstraction of these elements. In this paper, we present the work of an SDN approach for designing and implementing a Rate/Link Adaptation (RA) framework for wireless local area networks (WLAN). The framework provides support for real-time RA applications and flexibility to satisfy various degrees of Quality of Service (QoS) or Quality of Experience (QoE) requirements. We implement the proposed framework as an extension to the Wireless Open-Access Research Platform (WARP), an FPGA based Software-Defined Radio (SDR) platform, with evaluation results indicating the feasibility of using SDN-RA under the stringent time constraints posed by the WLAN. To demonstrate the effectiveness of decoupling rate decision functions from the underlying wireless interface card and to highlight its applicability for a diverse set of scenarios, we present a use case deployed over the framework focusing on rate adaptation for individual traffic, and display optimization in different aspects, such as the reduction transmission errors.

**Keywords**— *Rate/Link adaptation, Software-Defined Networking, SDN agent, IEEE 802.11, Software-Defined Radio*

## I. INTRODUCTION

Wireless local area networks (WLANs) have become more and more popular and ubiquitous due to their ability to provide unrestricted connectivity and mobile access. The family of IEEE 802.11 standards, commonly known as Wi-Fi, has been widely used in residential, commercial and industrial environments as well as effectively on all devices, including access points (APs), mobile phones, laptops, TVs, etc.

The IEEE 802.11 wireless media standard supports multiple data bit-rates at the physical layer (PHY), where the device may transmit at a rate higher than the base rate if channel conditions so permit [1]. In order to choose the most appropriate transmission rate, various link adaptation algorithms at the MAC layer have been proposed. Rate or link adaptation is the selection of the optimal data transmission rate given the current wireless channel conditions.

In current RA design, the RA decision and action are highly coupled with the wireless network interface card (NIC), a fact that results in following shortcomings:

- The instantaneous received signal power in wireless communication systems is affected by many factors (signal fading due to distance, frame collision from simultaneous transmissions, and interference from other sources). Therefore, it is difficult to correctly determine or estimate the wireless channel

conditions by using fixed metric(s). Currently, RA always uses the same metric(s) in channel estimation.

- Limitations of the PHY/MAC implementations in commodity wireless cards result in that only a limited set of MCSs tries to map almost unlimited wireless channel conditions, which in turn causes an imperfect match between the selected data rate and the underlying wireless channel. Generally speaking, no RA algorithm can work perfectly in all wireless channel conditions.
- Different applications have different Quality of Service (QoS) or Quality of Experience (QoE) requirements. For example, online gaming, and high-definition multimedia streaming demand higher throughput, while critical services and infrastructure control may require high reliability and low latency. Since there is a trade-off among different performance metrics, it is difficult to design a RA algorithm that can meet all possible application requirements.

Therefore, the next generation of WLANs demands a new rate control mechanism with more flexibility, scalability and effectiveness, which serves as the main motivation of this work.

In the Software-Defined Networking (SDN) paradigm [2], by separating the control plane from the data plane, the controller makes the network forwarding decisions and directly controls the network to achieve desired operating behaviours, hence enhancing the network control programmability. Therefore, SDN is a natural enabler for developing innovative rate control mechanisms that decouples RA decision functions from underlying NICs in a simplified and improved way.

In this paper, we develop a framework that, to the best of our knowledge, is the first Software-Defined Rate Adaptation (SDN-RA) framework for WLANs. The framework incorporates SDN agent with its APIs to separate the control and data planes. An SDN agent is a software element bridging the SDN controller and any legacy wireless network elements by providing the abstraction of these elements. The SDN agent design and implementation take into consideration the need to make real-time RA decisions feasible. The benefits of this SDN-RA idea are two-fold: i) the framework offers a great degree of flexibility to dynamically satisfy different QoS/QoE requirements; and ii) building RA algorithms in form of SDN control applications at the SDN controller side provides the ability of updating RA functions on the fly as well as easier deployment and evolution.

In summary, our main contributions in this work are:

- Design and development of an SDN-RA framework, a novel approach for flexible WLAN RA implementations. SDN agent APIs facilitate WLAN AP integration with the SDN domain by providing abstraction of WLAN resources.
- An implementation of the proposed framework using the WARP 802.11 platform, a highly configurable SDR system. The prototype has been shown to be efficient for real-time rate adaptation operations, after we thoroughly characterize the performance of the framework under different network conditions.
- Provision of proof-of-concept use case to showcase the capability of SDN-RA approach, including applying different RA strategies for various traffic patterns and under different QoS requirements.

The rest of the paper is organized as follows. Section II provides a brief background information on SDN, WARP platform and rate adaptation in WLAN. Section III and IV describe the design of the SDN-RA framework and the implementation on the WARP platform. In Section V, experiments are presented to show the capability of SDN-RA approach, followed by Section VI with conclusion and future work.

## II. BACKGROUND INFORMATION AND RELATED WORKS

In this section, we present the basic concepts of SDN and WLAN rate adaptations as well as a brief description of related work.

According to the Open Networking Foundation (ONF) white paper [2], the SDN architecture is comprised of the application, control and data plane. The data plane consists of network elements with the Control-Data-Plane Interface (CDPI) agent to expose the capabilities of SDN datapaths to the controller. In order to do this, the CDPI agent is responsible for carrying out the commands of the controller and notifying the controller of events that are specified by the controller. It is also noted that, in the control plane, there are agents called Northbound Interfaces (NBI) agents, which are interfaces between SDN controllers and SDN applications. In this paper, we focus on the agents in the data plane, while the NBI agents are out of scope of this paper.

In recent years, a significant amount of research work has been done to integrate SDN principles with WLAN. Odin [3] is an SDN Wi-Fi network architecture based on the concept of light virtual AP (LVAP), which separates states of each station association from physical APs. LVAPs are located on Odin agents that operate on APs and are implemented in the Click Modular Router [4]. In [5], an SDN framework is presented to extend the basic concept of SDN to WLANs. *AEtherFlow* is implemented in CPqD SoftSwitch [6], which is installed on OpenWRT [7]. The authors of [5] also present a cross-layer architecture, *CrossFlow* [8], using the principles of SDR and basic building blocks of SDN. It has been implemented using GUN radio [9] on a Universal Software Radio Peripheral (USRP) platform. While the aforementioned works address key network issues of traffic and user mobility management, and enable a degree of virtualization in WLANs, none of them addresses MAC and PHY radio resource management,

such as rate adaptation, a vital aspect of the WLAN and the focus of our work.

An RA algorithm consists of assessing channel conditions and accordingly adjusting the data rate. Rate adaptation algorithms can be classified into two categories: SNR based or packet transmission (loss) based [10][11][12]. In the SNR based rate adaptation algorithms, the received signal strength information (RSSI) is used as an indication of link quality, and then a transmission rate is selected based on the average or instantaneous RSSI from a predetermined SNR-rate table. Receiver Base Rate Fallback (RBRF) [10] is a typical example of such algorithms. In the packet retransmission based rate adaptation algorithms, the transmitting terminal counts the outcome (either succeeded or failed) of each transmission attempt. Based on the packet transmissions history, the transmitting rate can be adaptively adjusted. Auto Rate Fallback (ARF) is the first documented bit-rate selection algorithm [11].

The common and key limitation of the above mentioned RA works, which is a driving factor behind our work, is that they only focus on the throughput optimization and only achieve optimal throughput in a particular radio frequency (RF) environment [13]. Moreover, they are all closely-coupled with hardware NICs, however NICs may only be able to use one RA for all types of traffics under all kinds of network conditions at any given point in time. This rigid implementation is prone to performance degradation in the dynamically changing wireless environment and is difficult to support different QoS or QoE requirements.

The Wireless open-Access Research Platform (WARP) [14] is a highly programmable, scalable and extensible wireless platform. The platform is comprised of both custom hardware and field-programmable gate array (FPGA) implementations of key communications blocks. The Mango Communications 802.11 Reference Design [15] is a complete FPGA implementation of IEEE 802.11 MAC and PHY. In this paper, we focus on the SDN agent designed and developed for 802.11 APs based on the WARP reference design.

Both the hardware specifications and algorithm implementations are freely available via the online open-access repository. A complete FPGA architecture of the WARP 802.11 reference design can be found in [15].

## III. SYSTEM OVERVIEW

This section describes a high-level overview of an SDN-RA framework, the key contribution of this paper. To correspond to its current implementation, we use the IEEE 802.11 WLAN terminology of AP and STA for access point and mobile station respectively. Also due to the limit of pages, we only focus on functions related to RA. Fig. 1 provides a high-level schematic of the proposed framework, which consists of four main components: RA applications, a control plane, an SDN agent and a data plane. In the application plane, each RA module represents an RA algorithm implementation, which makes rate decisions based on the channel statistics obtained from the agent and its own decision logic. The RA management module is mainly responsible for associating an RA with a traffic flow on the basis of various QoS or QoE requirements. The SDN controller is the main component of the control plane, linking the RA applications and the SDN agents. In terms of RA, the

controller is in charge of delivering the RA decisions from RA applications to SDN agents and providing RA applications with an abstract view of the network, for example through statistics and events. The control plane and data plane separation is enabled by the SDN agent and its API. Each AP has its own SDN agent, while the SDN controller connects multiple SDN agents.

The main functionality of the SDN agent is to facilitate communications between the SDN controller and WLAN AP by establishing a two-way channel between them. In one direction, the agent receives and processes rate control instructions from the SDN controller and passes them to the data plane, while in the other direction the agent constructs and sends response to the SDN controller along with statistics of the underlying wireless networks. The SDN agent API works as the southbound API with the data plane on one side and the SDN control plane on the other side. As the design is flexible, the SDN agent functionalities can be implemented either in the control plane, or in the data plane, or even split between them based on the deployment scenario. Since our design takes into time-critical RA operations into account, ideally the communication channels should be high-bandwidth and low-latency, in order to fully utilize the capability of our framework.

The northbound API allows RA applications to change data bit-rates of APs in the data plane based on the network statistics and events exposed by the data plane. The WLAN AP network element (NE) is a logical entity in the data plane, representing the MAC and PHY functionalities of IEEE 802.11 access point. One of the main function modules in AP NE is the forwarding/traffic processing engines that decide how the traffic flows are to be handled. With regard to RA, this is where the SDN agent applies the RA decision made by RA application to the AP. In other words, the RA application affects the behaviour of 802.11 AP NE, in the form of changing data bit-rate, by the SDN agent.

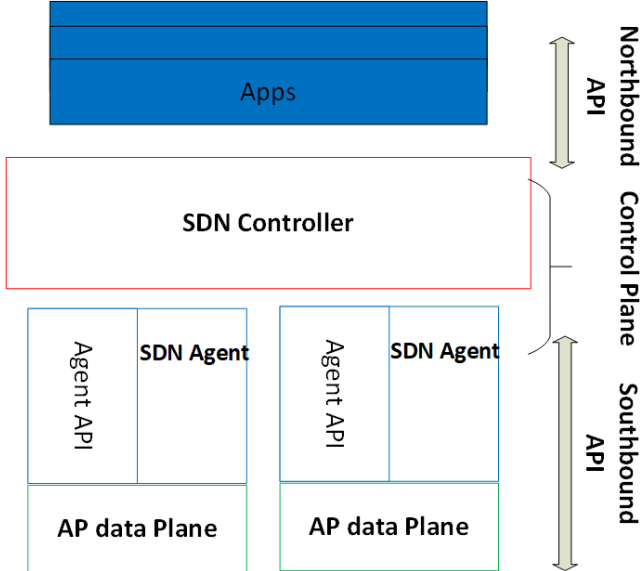


Fig. 1 High-level schematic of SDN-RA framework

It is noticed that Fig. 1. does not include a WLAN STA, which reflects the fact that the agent only operates at the AP side. Our approach is on par with the contemporary solutions. Besides, the dominant traffic in current WLAN is the downlink traffic from the AP to STAs.

## IV. SYSTEM DESIGN AND IMPLEMENTATION

### A. Design Challenges

Based on the discussion in the previous sections, two main challenges are identified in order to design an effective SDN-RA framework:

- Decoupling the RA control from the data plane in a clean and programmable way.
- Taking time critical RA operations into consideration when deploying RA applications over the controller.

In the following subsections, we present solutions to these challenges through a detailed description of each component of the framework as well as their implementation details over the WARP platform. We describe our design in a top-to-bottom fashion, starting with the RA implementations over the controller. Note we use Ryu SDN controller as the control plane in our framework and build our RA applications using the northbound API provided by Ryu. In this manuscript, we omit the control plane implementation detail, but for the interested reader, more information can be found in [16].

### B. RA Application Implementation

SampleRate is one of the state-of-the-art RA algorithms and has been widely used in Linux 802.11 drivers. SampleRate tries to maximize overall throughput through transmitting frames with the smallest average transmission time [17]. The most innovative design of this RA is that it periodically sends probing frames at new bit-rates in an attempt to find higher throughput. In 802.11, if a frame has been retransmitted three times and the transmitter does not receive the acknowledgement from the receiver, the frame is dropped and this is counted as a failure. If one bit-rate results in four consecutive failures during the transmission, SampleRate will use a lower bit-rate.

For practical reasons, we use frame error rate (FER) to replace the four successive failures as the rate change threshold. The default FER threshold is 10%, the maximum error rate allowed in 802.11 standard [1]. For every 1 second, the SampleRate retrieves transmission statistics from the AP and calculates the FER of each bit-rate used in this period. Based on the latest FER values, SampleRate decides whether or not to change to a new bit-rate. If it decides to change the bit-rate, SampleRate passes the new bit-rate to the RA management module, where the bit-rate is assigned to traffics according to their QoS or QoE requirements. Those QoS and QoE requirements are provided by other applications through the Ryu controller's REST API. The RA management module passes RA instructions to the SDN controller, which in turn conveys the messages to the corresponding SDN agent.

RA applications are currently implemented in Python and run as components on the Ryu SDN controller. At this stage, we only implement SampleRate, while other RAs can be implemented with the same approach.

### C. SDN WLAN Agent and API

Most legacy RA protocols of WLANs are implemented as a functionality on the MAC layer and can be divided into two parts: the rate control part that makes decisions for RF

radio and the action part that is responsible for applying those decisions. As we described in the previous subsection, in SDN-RA, the RA control logic has been delegated to the control plane over the controller. As a result, the APs only deal with traffic flows in the data plane by applying the rate decisions. To separate the control plane from the data plane, we introduce the SDN agent with its API, which defines a set of functions that serve as a southbound API and facilitate the Software-Defined control of APs. With these functions, the control plane can interact with the data plane. In terms of RA, the control plane can 1) request and obtain transmission statistics, such as the number of retries and the number of frames successfully delivered; 2) issue commands to apply rate decisions.

Instead of using a conventional OpenFlow-based one, we build a customized southbound protocol by extending the message system of the WARP experiments framework [15]. The main reason is that RA operations are highly associated with radio resource control, while OpenFlow protocols are mainly designed for the wired network domain. A command message is basically constructed with a message ID and a list of arguments. The RA instructions are supplied within a list of command arguments constructed as: [ARG0, ARG1, ..., ARGn] and the arguments would contain the following fields:

- Matching fields: fields in the IP header
- Action types: write, update, or read
- Action parameter pairs: <key, value>

For each message, a message handler is implemented in the SDN agent to execute the instructions issued by the SDN controller. Once the SDN agent interprets the command from the SDN controller, it will create a filter in forwarding engines in AP NE, right at the start of the Transmit (Tx) datapath. The filter contains the match condition and actions stated in the command message.

For practical reasons, we implement the SDN WLAN agent in the data plane on the WARP platform. The agent is developed in C and sits right on top of higher MAC in 802.11 AP, which means that the added agent functions will not affect the performance of lower-layer Tx/Rx tasks. The physical connection from the agent to the controller is via a Gigabit Ethernet cable, which is proved to be adequate in the system evaluation discussed in section V.

#### D. WLAN AP in Data Plane

One of the major advantages of the SDN concept is the flow-control. In order to demonstrate the capability of applying rate decision at per flow level as well as a proof-of-concept, we develop a simple and flow-table-like forwarding engine in AP NE in form of a set of filters. Those filters can match the type of service, protocol, source and destination IP addresses fields of the IP header. There is no need to develop a fully functional OpenFlow-switch in the data plane in our framework and this is for two reasons: 1) routing is not a main task for WLAN APs; 2) OpenFlow specification does not define the control in the wireless domain.

When a new data frame enters the Wi-Fi NE, the WARP framework will create metadata attached to this new frame. The metadata contains the description of how this frame should be dealt with when it is going through each stage of the 802.11 datapath. The data bit-rate is one of the

parameters in the metadata. The new data frame will enter a chain of filters created by the SDN agent. When it matches a specified filter, the bit-rate in its metadata will be modified accordingly, indicating how this frame will be transmitted through the 802.11 radio.

Fig. 2 depicts the architecture view of SDN-RA implementation.

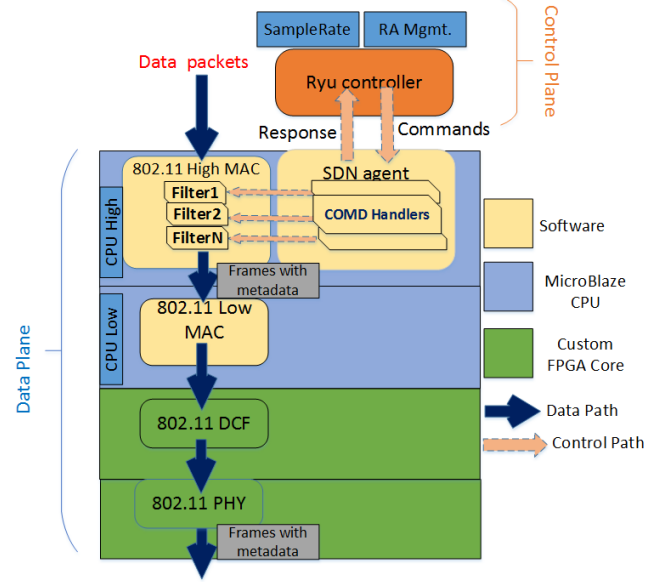


Fig. 2. Implementation of SDN-RA framework

## V. SYSTEM EVALUATION

### A. Experimental Setup

Our test-bed uses a simple network topology, shown in Fig. 3. It is mainly comprised of a PC hosting the SDN controller and traffic generator, one WARP board acting as AP and one off-the-shelf 802.11 client (STA). As described in Section IV, an SDN WLAN agent is implemented on the WARP board. There are two separate Ethernet cables between the PC and WARP board, one for the control path between the Ryu SDN controller and SDN agent, and the other for the data path between the traffic generator and AP. We also use a PROPSIM C8 channel emulator in some experiments. The 802.11 channel models supported by the emulator are listed below.

Table 1 Wi-Fi MODELS

Model Name	Description
A	a flat fading channel
B	residential environment
C	small office environment
D	typical office environment
E	large office environment
F	large indoor and outdoor environment

### B. Scalability

In this subsection, we evaluate whether the SDN-RA framework can cope with the increase of number of APs. One essential metric judging the scalability is the network



overhead. In this case, we measure the signalling overhead between the SDN controller and the agent. In RA application, the main source of the agent-to-controller overhead is the periodic transmission statistics reports followed by the controller-agent synchronization messages. Although SampleRate only uses FER and Tx time information, the report also contains RSSI, throughput, Tx queue status and loss rate, which are important metrics in RA. For every one second, the overhead is 440 bytes for one AP. In the case of the controller-to-agent signalling, the overhead comes almost from RA command messages, which is 1140 bytes for one AP in one second. This shows that the communication channel implementation in our framework is a lightweight design and can support a large number of APs.

One important thing to notice is that the sampling rate in RA is a major factor for the agent-to-controller overhead. Although one second is enough for most of RAs, higher sampling rates may be used in some cases, thus increasing overhead.

### C. Latency Impact on Control Channel

One crucial feature for time-critical controls, such as, SDN-RA is the impact of the control channel latency between the agent and the controller. To test its effect we use *netem* tool [18] to add various network latency in the control channel and measure the STA downlink throughput for various network channel configurations created by C8 emulator. The normalized throughput is the one measured with network delay divided by that measured when there is no network delay, which is shown in Fig. 4.

As we can see, the throughput generally decreases when network latency is introduced. However, the change is very small, ranging from 1% to 6%. This indicates that SampleRate is still able to make correct rate decisions even when it has to rely on outdated channel statistics reports, and the decision is still valid when it arrives to the agent. Model F has the lowest throughput and the largest decrease, as it has the most dynamic channel conditions, hence being affected the most.

### D. Use Case

In this subsection, we demonstrate the usefulness of the SDN-RA framework by presenting a use case that SDN-RA can enable. As we discussed in previous sections, different traffic types have various QoS requirements and WLAN has to support all types of services. However, current RA algorithms are mainly focusing on maximizing the data bit-rate, hence only optimizing throughput-driven applications. Because the RA control is closely coupled with NIC in the data plane, the AP is only able to use one RA algorithm for all types of traffics. In this experiment, we demonstrate that our approach allows to use different RA strategies for different traffics according to their QoS or QoE requirements.

As a proof-of-concept, we implement a variation of SampleRate with a smaller error rate threshold (5%). This forces SampleRate to choose the bit-rate that makes less transmission errors thus achieving the reliability. The RA management module assigns the normal SampleRate (10% error threshold) to UDP traffic, as it is the main transportation protocol used for the throughput-driven applications.

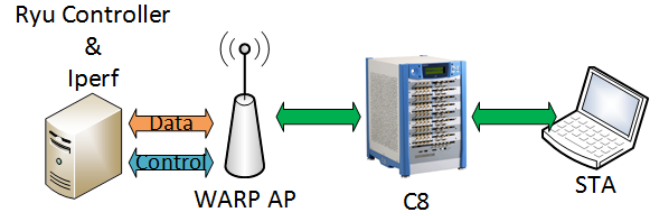


Fig. 3. Experiment topology

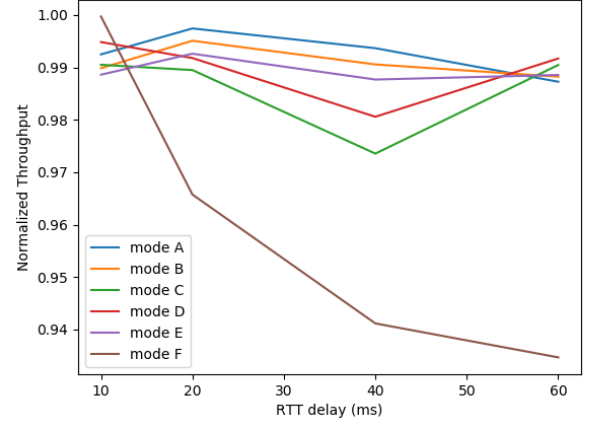


Fig. 4. Normalized throughput vs. RTT time

On the other hand, the RA management assigns the SampleRate with 5% error rate threshold to TCP traffic, as most applications using TCP require reliable delivery of data stream. In order to do this, the RA management sets the protocol field of the IP header in the matching field of RA instruction, while setting the rate decision of the corresponding SampleRate strategy in the action field. By doing this, the RA management can constantly update the SDN agent with the latest rate decision. The RA management can also change the RA strategy for a traffic on the fly.

When the SDN agent receives the RA command message, it creates a filter with two match-and-action rules in the AP on the data plane. Therefore, when a UDP traffic enters this filter and matches the rule, the data bit-rate in the metadata of each frame of this UDP traffic will be updated. The same operation also works for TCP traffic. Finally, at the 802.11 radio, the frame, either UDP or TCP, will be transmitted with the bit-rate set in its metadata.

Two traffic flows, one UDP and one TCP, are generated from the AP to the STA. The flows can be differentiated by the protocol field in their IP headers. C8 emulator is used to create various Wi-Fi channels.

Fig. 5 depicts the average FER reported by the SDN agent during the experiment, showing that the error rate of TCP traffic is always smaller than that of UDP traffic in all channel models except when in model A, where UDP error rate is slightly smaller than that of TCP traffic. We can also observe that TCP error rate is almost flat in model A, B, C and D, while UDP error rate largely fluctuates. One more thing to notice is that the error rate of TCP traffic is always below 5%, since it uses bit-rate with less transmission errors.

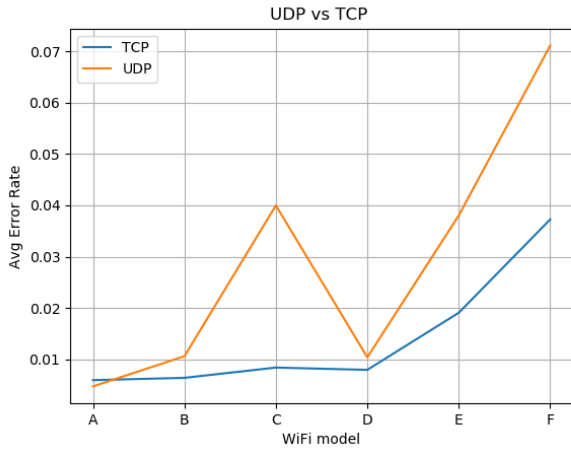


Fig. 5. Performance of UDP and TCP vs. various WiFi models

## VI. CONCLUSION

In this paper, we presented an SDN-RA framework, a flexible and programmable platform for the implementation of RA algorithms in WLANs. The proposed approach enables the separation of the control plane from the data plane through a customized agent and its APIs, while providing real-time RA operations in WLANs. The implementation of our framework over the WARP platform and our evaluation results confirm the feasibility of its deployment even when RA operations are considered to be time-critical. The effectiveness of this framework is highlighted by a use case, focusing on reducing transmission errors. The proposed approach provides a greater degree of flexibility and programmability than current RA solutions and opens a new door for developing innovative RA mechanisms for agent-based APs or base stations.

The main features of our SDN-RA agent framework are listed below:

- 1) Different rate adaptation strategies can be applied to different applications based on their own QoS or QoE requirements, while different rate adaptation strategies can be selected according to underlying channel environments.
- 2) There is no need to modify the SDN elements and 802.11 protocols.
- 3) Real-time control of data bit-rate of WLAN AP via SDN controller can be achieved without interfering the real-time operation of Wi-Fi interfaces.

Here we outline several extensions for future work on our SDN-RA framework. Firstly, more state-of-the-art RA algorithms can be implemented as SDN applications within the SDN controller, which gives the RA module more selection options. Secondly, we would like to explore the ideas of applying machine-learning strategies in RA framework in a bid to design more intelligent rate selection algorithms. Lastly, we would also extend our framework to LTE/LTE-A networks by developing LTE agent. The results of our work will provide a promising way of convergence of all RA/LA frameworks for all kinds of wireless networks.

## ACKNOWLEDGMENT

The authors would like to thank Mr. Junze Xu for the implementation of SampleRate on SDN controller and Dr. John Harris for the help on WARP platform. This work is supported by EPSRC grant EP/L020009/1: Towards Ultimate Convergence of All Networks (TOUCAN).

## REFERENCES

- [1] IEEE 802.11ac-2013 "IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications—Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz" 2013.
- [2] Open Networking Foundation. "Software-Defined Networking: The New Norm for Networks," 2012. [Online]. Available: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>
- [3] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise wlangs with odin," in HotSDN, August 2012, pp. 115–120.
- [4] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, M. Frans Kaashoek, "The click modular router," ACM Transactions on Computer Systems (TOCS), v.18 n.3, p.263-297, Aug. 2000
- [5] M. Yan, J. Casey, P. Shome, A. Sprintson, and A. Sutton, "Aetherflow: Principled wireless support in sdn," in IEEE 23rd International Conference on Network Protocols (ICNP), 2015, pp. 432–437.
- [6] "CPqD SoftSwitch." [Online]. Available: <https://github.com/CPqD/ofsoftswitch13>
- [7] "OpenWRT." [Online]. Available: <https://openwrt.org>
- [8] P. Shome, M. Yan, S. M. Najafabad, N. Mastronarde, A. Sprintson, "CrossFlow: A cross-layer architecture for SDR using SDN principles", Proceedings of the IEEE NFV-SDN Conference, pp. 37-39, November 2015.
- [9] "GNU Radio". [Online]. Available: <http://enuradio.org/redmine/projects/anuradio/wiki>.
- [10] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," ACM MobiCom 2001, Rome, Italy, 2001.
- [11] A. Kameman and L. Montban, "WaveLAN II: A high-performance wireless LAN for the unlicensed band," Bell labs Technical Journal, pages 118-133, 1997.
- [12] M. Lacage, M. hossein and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," IEEE MSWiM, October 2004.
- [13] S. Biaz S. Wu "Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison" Proc. IEEE ISCC pp. 130-136 Jul. 2008.
- [14] K. Amiri, Y. Sun, P. Murphy, C. Hunter, J. R. Cavallaro, and A. Sabharwal, "WARP, a unified wireless network testbed for education and research," in Proc. IEEE Int. Conf. Microelectron. Syst. Edu., Jun. 2007, pp. 53–54
- [15] "WARP project". [Online]. Available: <https://warpproject.org>.
- [16] "Ryu SDN Framework." [Online]. Available: <http://osrg.github.io/ryu>.
- [17] J. C. Bicket, "Bit-rate selection in wireless networks," Massachusetts Institute of Technology, 2005.
- [18] <https://wiki.linuxfoundation.org/networking/netem>